Dirt Classification using CNN and Streamlit

# Objective

To build an image classification system that categorizes images into 8 classes such as Blur, Clear, Dust, Foggy, Rainy, Snow, StickerAndSplodges, and Tinted. The system also computes visibility metrics and classifies the severity of dirt using RAG status indicators.

# Process Overview

1. A CNN model is trained using Keras and TensorFlow with augmented image data from 8 categories.  
2. The trained model is integrated into a Streamlit application.  
3. The Streamlit app allows the user to upload or select a folder of test images.  
4. For each image, the model predicts the class and calculates dirt percentage, weather-related dirt percentage, and visibility metrics.  
5. Results are displayed in a user-friendly interface and can be downloaded as a CSV.

# Detailed Process Flow

1. **Image Acquisition**  
   Images are collected from a specified folder path. These images are expected to be of varying quality and may contain visual obstructions such as Blur, Clear, Dust, Foggy, Rainy, Snow, Sticker & Splodges, and Tinted.
2. **Preprocessing**  
   Each image is resized to 128×128 pixels and normalized to ensure consistent input to the CNN model. A grayscale version is also created for auxiliary tasks like visibility scoring and incident detection.
3. **Obstruction Detection**
   * The preprocessed image is passed through a trained CNN model which classifies it into one of eight categories: Blur, Clear, Dust, Foggy, Rainy, Snow, StickerAndSplodges, or Tinted.
   * Confidence scores for each category are calculated.
   * Dirt and weather-based confidence percentages are derived.
   * A RAG (Red-Amber-Green) status is assigned based on dirt scores.
   * Additional analysis like incident vs. degradation detection is performed using grayscale image contrast and dark pixel ratios.
4. **Result Compilation**  
   For each image, a result dictionary is prepared containing:
   * Predicted label and confidence
   * Dirt %, Clear %, Weather Dirt %
   * RAG status
   * Type (Incident/Degradation/Unknown)
   * Visibility score and contributing metrics (sharpness, brightness, contrast)
   * Confidence scores for all classes
5. **Reporting**
   * Results are displayed using a Streamlit interface with image previews and expandable sections for detailed analysis.
   * A summary table of all image results is generated.
   * Users can download the entire report as a CSV file.

# Tools and Technologies

* **Programming Language**: Python
* **Libraries**:
  + OpenCV: Image processing
  + TensorFlow/Keras: Model development
  + Pandas: Data manipulation and Excel export
  + Streamlit: UI Path

# Appendix

